

# THAM KHẢO NHANH NPM

Quản lý package, scripts, versioning, publishing, workspaces

## Cài Đặt

### Cài Đặt npm & Node

```
node -v && npm -v # check installed versions
npm install -g npm@latest # update npm itself
nvm install --lts # install Node LTS via nvm
nvm use 20 # switch to Node 20
```

### Lệnh Cài Đặt

```
npm install # Cài tất cả dependencies từ package.json
npm install pkg # Thêm package vào dependency
npm install -D pkg # Thêm package vào devDependency
npm install -g pkg # Cài package toàn cục
npm install pkg@2.1.0 # Cài phiên bản cụ thể
npm ci # Cài sạch từ lock file (CI/CD)
npm uninstall pkg # Gỡ bỏ package
```

## Quản Lý Package

### Quản Lý Packages

```
npm ls # list installed packages
npm --depth=0 # top-level only
npm outdated # check for newer versions
npm update # update within semver range
npm audit # check for vulnerabilities
```

### Lệnh Quản Lý

```
npm ls # Liệt kê packages đã cài đang cây
npm outdated # Hiển thị packages có phiên bản mới hơn
npm update [pkg] # Cập nhật packages trong phạm vi semver
npm audit # Kiểm tra lỗ hổng bảo mật trong dependencies
npm audit fix # Tự động sửa lỗ hổng
npm prune # Xóa các packages không cần thiết
npm dedupe # Làm phẳng dependency tree để giảm trùng lặp
```

## Scripts

### Chạy Scripts

```
npm run build # run "build" script
npm test # shortcut for "test" script
npm start # shortcut for "start" script
npm run lint -- --fix # pass args to script
npm run dev & # run in background
```

### Script Lifecycle

```
npm test / npm t # Chạy `scripts.test`
npm start # Chạy `scripts.start`
npm run <name> # Chạy bất kỳ script tùy chỉnh nào
pre<name> # Tự động chạy trước <name>
post<name> # Tự động chạy sau <name>
npm run # Liệt kê tất cả scripts có sẵn
```

## package.json

### Khởi Tạo & Trường

```
npm init # interactive setup
npm init -y # accept all defaults
npm pkg set name="my-app" # set a field
npm pkg get version # read a field
```

### Trường Quan Trọng

```
name # Tên package (chữ thường, không dấu cách)
version # Phiên bản hiện tại (semver: major.minor.patch)
main # Entry point cho CommonJS (require)
module # Entry point cho ES modules (bundlers)
type # "module" cho ESM, "commonjs" cho CJS (mặc định)
scripts # Lệnh được đặt tên ("build", "test", "start", v.v.)
dependencies # Dependencies sản xuất
devDependencies # Dependencies chỉ dùng khi phát triển
engines # Yêu cầu phiên bản Node/npm
```

## Versioning

### Lệnh Version

```
npm version patch # 1.0.0 -> 1.0.1
npm version minor # 1.0.1 -> 1.1.0
npm version major # 1.1.0 -> 2.0.0
npm version 3.2.1 # set explicit version
npm version prerelease --preid=beta # 1.0.0-beta.0
```

### Khoảng Semver

```
~1.2.3 # Tương thích: >=1.2.3 <2.0.0 (mặc định)
~1.2.3 # Patch-level: >=1.2.3 <1.3.0
1.2.3 # Chỉ phiên bản chính xác này
>=1.0.0 <2.0.0 # Khoảng tương minh
* # Bất kỳ phiên bản nào
1.x / 1.2.x # Khoảng wildcard
latest # Tag phiên bản mới nhất đã publish
```

## Publishing

### Quy Trình Publish

```
npm login # authenticate to registry
npm publish # publish public package
npm publish --access public # scoped package as public
npm unpublish pkg@1.0.0 # remove specific version
npm deprecate pkg@<2 "Use v2+" # deprecate old versions
```

### Tham Khảo Publish

```
npm login # Xác thực với npm registry
npm publish # Publish package lên registry
npm pack # Tạo tarball mà không publish
npm unpublish # Xóa phiên bản đã publish (trong 72h)
npm deprecate # Đánh dấu các phiên bản là deprecated
npmignore # File cần loại trừ khỏi package đã publish
files (package.json) # Danh sách file được phép đưa vào package
```

## Workspaces

### Lệnh Workspace

```
npm init -w packages/core # create workspace
npm install -w packages/core lodash # install in workspace
npm run build -w workspaces # run in all workspaces
npm run test -w packages/api # run in specific workspace
npm ls -w workspaces # list workspace deps
```

### Cấu Hình Workspace

```
workspaces (package.json) # Mảng workspace globs:
# "packages/*"
-w / --workspace # Nhắm đến workspace cụ thể
--workspaces # Chạy lệnh trên tất cả workspaces
--include-workspace-root # Bao gồm package root trong workspace
```

```
npm install (root) # Cài tất cả workspace dependencies
Hoisting # Shared deps được nâng lên root node_modules
```

## npmx

### Chạy Với npmx

```
npmx create-react-app my-app # run without installing
npmx tsc -init # run local or remote bin
npmx -p typescript tsc file.ts # specify package explicitly
npmx --yes create-next-app # skip install prompt
npmx node@18 -e "console.log('hi')" # run with specific Node
```

### Tùy Chọn npmx

```
npmx cmd # Chạy cmd từ node_modules/.bin hoặc remote
npmx -p pkg cmd # Cài pkg, rồi chạy cmd
npmx --yes cmd # Tự xác nhận cài đặt
npmx --no cmd # Từ chối cài đặt — thất bại nếu không có local
npmx -c 'cmd' # Chạy shell command với PATH của npm
npmx node@ver # Chạy phiên bản Node.js cụ thể
```

## Cấu Hình

### Lệnh Cấu Hình

```
npm config list # show current config
npm config set registry https://r.npmjs.com/
npm config set init-author-name "Name"
npm config get prefix # global install path
npm config delete key # remove a config value
```

### Tham Khảo Cấu Hình

```
npmrc (project) # File cấu hình theo project
./npmrc # File cấu hình theo user
registry # URL của package registry
save-exact # true để ghi phiên bản chính xác khi cài
engine-strict # true để bắt buộc trường engines
fund # false để ẩn thông báo tài trợ
audit # false để bỏ qua audit khi cài
```

## Mẫu Phổ Biến

### One-Liners

```
npm ls --depth=0 --json | jq '.dependencies | keys[]'
npm outdated --long # show type and homepage
npm cache clean --force # clear npm cache
npm explain pkg # why is pkg installed?
npm exec -- envinfo --system # system info for bug reports
```

### Công Thức Hữu Dụng

```
Chỉ lock file # npm ci — cài sạch từ package-lock.json
Kiểm tra license # npx license-checker --summary
Tìm deps không dùng # npx depcheck
Kích thước bundle # npx bundlephobia-cli pkg — kiểm tra kích thước
Nâng cấp tất cả # npx npm-check-updates -u && npm install
Registry nội bộ # npx verdaccio — chạy private registry
```